UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/823,009	04/13/2004	Michael W. Shapiro	03226/369001; SUN040527	4684

32615 7590 11/07/2008

OSHA LIANG L.L.P./SUN
TWO HOUSTON CENTER
909 FANNIN, SUITE 3500
HOUSTON, TX 77010

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

NOTIFICATION DATE	DELIVERY MODE
11/07/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docketing@oshaliang.com
lord@oshaliang.com
hathaway@oshaliang.com

Office Action Summary	Application No.	Applicant(s)	
	10/823,009	SHAPIRO ET AL.	
	Examiner	Art Unit	
	TUAN A. VU	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 June 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-8 and 10-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-8 and 10-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 6/23/08.

As indicated in Applicant's response, claims 1-2, 4, 7, 13, 23 have been amended, and claim 9 canceled. Claims 1-8, 10-23 are pending in the office action.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-8, 10-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over John Murayama, "Performance Profiling Using TNF", July 2001, pg. 1-6 (hereinafter Murayama), in view of Berry et al, USPN: 6,728,955 (hereinafter Berry).

As per claim 1, Murayama discloses method for tracing an instrumented application, comprising:

loading the instrumented application into a kernel level (e.g. ... *inserted into the application program or kernel code ... TNF probes into the source code* - TNF overview, pg. 1, middle) to obtain a corresponding instrumented process (TNF execution trace - middle pg. 1);

registering a helper action (e.g. sec. **Instrumenting the Target**: ... interval around a *print* statement – pg. 2; **Interposition Libraries** – pg. 3... Example 3: *"interposed" tnf_unikt, v, v*) with a tracing framework;

tracing the instrumented process using the tracing framework (e.g. *probes ... to trace basic kernel events such as syscalls, I/O operations, page out* - TNF overview – pg. 1, middle), wherein tracing comprises

triggering a probe (e.g. **probes** … to trace basic kernel events such as syscalls, I/O operations, page out - TNF overview – pg. 1, middle; sec **Inserting Probes**, pg. 2) in the instrumented process;

determining whether the helper action is associated with the probe (e.g. set of libraries and utilities – TNF overview pg. 1; **Interposition Libraries** – pg. 3 - Note: setting interval between start tnf_probe and end tnf_probe – Example 3, pg. 3 - to call a libraries "interposed" code reads on determining existence of association of probe and "interposed"); and

performing the helper action if the helper action is associated with the probe (e.g. Example 3: Interposed, pg. 3; Example 1: *printf* - pg. 2).

Murayama does not explicitly disclose that the helper action is for obtaining a *stack trace* of the instrumented process, such that performing the helper action when associated with a probe is to obtain a *stack trace*. Using hooks like Murayama to record trace implicated with interrupt handling triggered on events regarding method calls and stack structures is disclosed in Berry (*interrupt handler trace hook* – Fig. 7; col. 11, bottom to top col. 12; trace hook … in response to event… method entry or exit … call stack record – col. 10 lines 33-42; event-based profiling … timer interrupt … trace facility - col. 12 lines 19-67; Fig. 8), based on which Berry uses Sun Java compiling environment to instrument Java processes for tracing threads in part at the kernel level (e.g. *kernel* – col. 26, lines 12-41) and teaches set up of probe (e.g. Fig. 7; col. 12 lines 19-67) to instrument virtual machine stacks (e.g. Fig. 6, Fig. 8-9, 10A-B; Fig. 12) to monitor calls at

runtime. Based on needs for monitoring *virtual address* changes in memory by Murayama (see Example 8, pg. 4) when initiating probe insertions to trace memory accesses similar to Berry, it would have been obvious for one skill in the art at the time the invention was made to implement Murayama's tracing so that probe/hooks setting would be based on predetermined events to associate or enable helper action (or hook-handler) to invoke helper function as to record *stack trace* as in Berry because of the need to collect dynamic methods invocation change information regarding address of methods as purported above in Murayama just as this is endeavored by Berry in obtaining proper dereferencing pointer or routine calls offset update (see col. 12 lines 28-60)

As per claim 3, Murayama discloses linking the helper action to a specially self-describing file format source file (e.g. self–describing file format – top para, pg 2) associated (e.g. Example 1, Example 2, Example 3, pg; 2-3 – refer to claim 2 for helper libraries) with the instrumented application; and using the TNF format to initiate execution by means of a *prex* instruction (Example 6-8, pg. 3-4). But Murayama does not explicilty disclose that such TNF source file is an initialization file. In view of Murayama's mentioning of linking by the compiler because of the libraries documented from MAN page and Probe documentation file (see NPROBE; TNF_PROBE(3tnf) - pg. 3 top) the suggestion of linking a original TNF source format within an intermediate stage with additional file suggests a initial file being (as a Sun system Make facility) intermediately combined with more compiler support files or header files known in Sun/Solaris environment (see *Solaris TNF facility* - Introduction, pg. 1). In light of this pre-execution status and nature of data defined in the TNF particular format, it would have been obvious for one skill in the art at the time the invention was made to implement the

programming language and special format file by Murayama (i.e. a self-describing format file) so that this is a starting format file for initializing or defining actions as taught by Murayama, such that when linked with the probe-related libraries, this *initialization* file would serve as input to the intermediate translation format as being normally performed by a Solaris-based linking process using more files as set forth above. One would be motivated to do so because Murayama's framework thus endeavored using Solaris TNF facility should allow special language to initially set actions/definitions (e.g. as in a header file, or a input specification script) prior to linking; that is, in order to provide further readjustment, having such initial set of data prior to linking by means of standard Solaris compiling process (e.g. using a Make script or header files), such initial setting in form of file structure (e.g. as suggested via a specialized format file (e.g. as in a special format TNF script -- Murayama: Example 1, Example 2, Example 3, pg; 2-3; *tnftrace* wrapper script – bottom pg 5) would support revising analysis after any tracing instance, and subsequent readjusting of parameter/setting (as by Murayama's tracing framework) for the probe definition or libraries association --as approached by Murayama's Solaris based compiler -- can be conveniently effectuated to help improving the process of tracing complex kernel behaviors.

As per claim 4, Murayama does not explicitly disclose wherein loading the instrumented application comprises triggering a hook in the initialization file to load the helper action into the kernel-level for use by the tracing framework. But in view of Murayama use of a special format source file to initialize how probes (Example 1, Example 2, Example 3, pg; 2-3; *tnftrace* wrapper script – bottom pg 5) are set to order to hook the instrumented process with the probe-related handlers and the rationale to use probe handler to perform stack tracing as in claim 1, the

use of initialization file to triggers the hooks would have been obvious based on the rationale set forth in claims 1 and 3.

As per claims 5-6, Murayama does not explicitly disclose wherein the helper action is stored in a process helper data structure, wherein the process helper data structure is associated with the instrumented process. But the storing of specification for probes and trigger points in a file entails a data structure. The above helper data structure would be an obvious variation of file to initialize how probe and helper actions are defined; hence would have been obvious in view of the initialization file as addressed in claim 3.

As per claims 2 and 7, Murayama (in view of claim 1) discloses obtaining the helper action associated with the instrumented application (e.g. *set of libraries and utilities* – TNF overview pg. 1; **Interposition Libraries** – pg. 3 – Note: existing libraries requires obtaining a handle when performing libraries dynamic linking); and generating a helper action associated with the instrumented application (e.g. Example 3: Interposed, pg. 3; Example 1: *printf* - pg. 2)

As per claim 8, Murayama does not explicitly disclose linking the helper action to an initialization file associated with the instrumented application. But based on the rationale as set forth in claim 3, and Murayama linking of the TNF file with the libraries and probe external documentation, the linking of helper action with the initialization file and instrumented application would have been obvious for the same reasons as set forth above.

As per claims 10-11, Murayama discloses wherein the helper action is stored in a process helper data structure (refer to claim 5 – Note: file inherently includes data structure to contain programmatic constructs or specification parameters); wherein the process helper data structure is associated with the instrumented process (refer to claim 6).

As per claim 12, Murayama discloses wherein performing the action associated with the probe further comprises: performing a probe action (e.g. **Interposition Libraries** – pg. 3 - Note: setting interval between start *tnf_probe* and end *tnf_probe* – Example 3, pg. 3 - to call a libraries "interposed" code reads on determining existence of association of probe and "interposed"; Example 5: *this probe enables the interposition library to trace*, pg. 3; Example 7, pg. 4) associated with the probe.

As per claim 13, Murayama discloses a system, comprising a processor configured to execute:

an instrumented process corresponding to an instrumented application comprising a probe (**probes** … to trace basic kernel events such as syscalls, I/O operations, page out - TNF overview – pg. 1, middle; sec **Inserting Probes**, pg. 2), wherein the probe is associated with an action (e.g. Example 3: Interposed, pg. 3; Example 1: *printf* - pg. 2);

a helper action associated with the instrumented application (sec. **Instrumenting the Target**: … interval around a *print* statement – pg. 2; **Interposition Libraries** – pg. 3... Example 3: *"interposed" tnf_unikt, v, v*); and

a tracing framework configured to trace the instrumented process (e.g. probes … to trace basic kernel events such as syscalls, I/O operations, page out - TNF overview – pg. 1, middle) corresponding to the instrumented application and to execute the helper action (e.g. e.g. Example 3: Interposed, pg. 3; Example 1: *printf* - pg. 2; refer to claim 12) if the action is associated with the helper action; and a storage device to store the trace of the instrumented process (Note: trace record inherently teaches storage device).

Murayama does not disclose tracing being configured to perform the helper action to obtain a *stack trace* when the action is associated with the helper action, and storage device to store the stack trace. But this limitation has been addressed in claim 1.

As per claim 14, Murayama discloses wherein the helper action is generated using implementation specific details associated with the instrumented application (e.g. **Instrumenting the Target**: … interval around a *print* statement – pg. 2; **Interposition Libraries** – pg. 3... Example 3: *"interposed" tnf_unikt, v, v)*.

As per claim 15, Murayama does not explicitly disclose wherein the implementation specific details comprise at least one selected from the group consisting of an instrumented application data structure and an instrumented application algorithm. But in view of instrumented process by which Murayama initialize a special language and file structure for specifying actions and probes as set forth in claim 3, and the script file to lay algorithm for linking to additional files, libraries or header files as contemplated in Murayama's via the use of Solaris linking facility to combine TNF specification with the actions support libraries (refer to claim 3), this instrumented data structure and application algorithm falls under the ambit of a initial file with structure to coordinate an algorithmic process for linking files and libraries or macros as well-known in a Solaris compiler (e.g. using a Make facility), and would have been obvious in view of the rationale for the obviousness established in claim 3.

As per claims 16-17, Murayama does not explicitly disclose wherein the instrumented application data structure comprises an application stack; wherein the application stack comprises either an interpreter stack or a virtual machine stack. Sun Virtual machine using Java compiler for instrumenting code was known practice at the time the invention was made. Berry

uses Sun Java compiling environment to instrument Java processes for tracing threads in part at the kernel level (e.g. *kernel* – col. 26, lines 12-41) and teaches set up of probe (e.g. Fig. 7) to instrument virtual machine stacks (e.g. Fig. 6, Fig. 8-9, 10A-B; Fig. 12) to monitor calls at runtime. Based on needs for monitoring *virtual address* changes in memory by Murayama (see Example 8, pg. 4) when initiating probe insertions to trace memory accesses similar to Berry, it would have been obvious for one skill in the art at the time the invention was made to implement Murayama's tracing , so that **if** Java is main language implementation --such as taught by Berry's tracing method-- for Murayama's compilation using a Solaris environment, the runtime call stack memory within Murayama's Solaris virtual environment can be set by application data structure (as taught by Berry) comprising construct representing a JVM stack or interpreter stack as mentioned above, for the same reasons that any virtual memory access contemplated during stack usage by program must be monitored to prevent sudden conflicts (refer to Fig. 8, 10A and related text by Berry)

As per claims 18-19, Murayama discloses wherein the action is a generic tracing action (Example 1: *printf* - pg. 2), wherein only the helper action is executed if the helper action and the generic tracing action are associated with the probe (refer to claim 12).

As per claim 20, refer to claims 18-19 for " wherein the helper action and the generic tracing action are executed if the helper action and the generic tracing action are associated with the probe".

As per claims 21-22, Murayama does not explicitly disclose wherein the helper action is stored in a process helper data structure; wherein the process helper data structure is associated with instrumented process. But this data structure has been addressed in claim 3 and claim 15.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claim 23 is rejected under 35 U.S.C. 102(e) as being anticipated by Berry et al, USPN: 6,728,955 (hereinafter Berry).

As per claim 23, Berry discloses network system (e.g. Fig. 1) having a plurality of nodes, comprising a processor configured to execute:

an instrumented process corresponding to an instrumented application comprising a probe, wherein the probe is associated with an action (e.g. hook 602- Fig. 6); a helper action associated with the instrumented application (e.g. handler - Fig. 7); and

a tracing framework configured to trace an instrumented process and to perform the helper action to obtain a stack trace for the instrumented process (col. 11, bottom to top col. 12; *trace hook ... in response to event... method entry or exit ... call stack record* – col. 10 lines 33-42; event-based profiling … timer interrupt … trace facility - col. 12 lines 19-67; Fig. 8) if the action is associated with the helper action (e.g. Fig. 7; col. 12 lines 19-67); and a storage device to store the stack trace (Note: record of stack trace inherently teaches storage device)

wherein the instrumented application executes on any one of the plurality of nodes (e.g. Fig. 1, 2B, 5; col. 6, lines 49-63; col. 7, lines 10-12), wherein the helper action is located on any one of the plurality of nodes (e.g. Fig. 7), and wherein the tracing framework executes on any

one of the plurality of nodes (e.g. Fig. 2B; tracing with stack unwinds – col. 6, lines 49-63; col. 7, lines 10-12).

Response to Arguments

6. Applicant's arguments filed 6/23/08 have been fully considered but they are moot or not persuasive. Following are the Examiner's observation in regard thereto.

35 USC § 102 Rejection:

(A) Applicants took the position that Murayama's TNF probes for profiling are not helper action to obtain stack trace (Appl. Rmrks, pg. 9 middle). The argument is based on the added limitation, thus would be moot in light of the current rejection.

(B) Applicants have submitted that Berry's reconstructing of call stack based on event does not disclose hooks used to obtain a record of the current state of the stack (Appl. Rmrks, pg. 10, middle). The language of the claim only recites probe-based obtaining a stack trace by performing an action helper, no more no less. Berry's hooks stand for probe and event-based with interrupt setting has been analogized to handler action to be performed when such hooks or probes are triggered. Berry teaches performing action as to record a stack trace, and this is fulfilling the language of the claim. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

35 USC § 103 Rejection:

(C) Applicants have submitted that Murayama's TNF cannot support stack trace as in claim 1; nor can Murayama teach initialization file of claim 4, with linking of libraries or Make Library

to a tracing framework (Appl. Rmrks pg. 12). Claim 4 has been addressed as obvious regarding the limitation 'initialization file', and Applicants fail to show how 'initialization file' as proferred in the 103 rationale would not have been proper based on the findings in Murayama; nor can the Applicants show how as claimed the 'initialization file' and the 'for use by the tracing framework' dictate a compelling relationship (emphasis added: how 'for use' and hook to load are coordinated?) so as to prevent the 103 rationale to fail to render this hook initiating structure/setting (or special format file used for loading as in the rejection) non-obvious or un-applicable. The Applicants' observation (Appl. Rmrks pg. 12 bottom pg. 13 top) such as to negate a possibility in Murayama (e.g. Make file cannot be used for a tracing framework?) without pointing factually how this cannot be possible is deemed not establishing how Murayama would not support creating of a special file to initiate a hook set up when hook or probe (for monitoring or profiling as in Murayama) already entails tracing or capturing of runtime data. Based on the lack of specifics in the claim and in view of the endeavor by Murayama, the feature 'initialization file' to support hook settings has been deemed obvious as set forth in the Rejection and the Applicants' allegation is not sufficient to overcome the rejection.

In view of the changes made to the claims and the adjusted grounds of rejection, the claims stand rejected as set forth in the Office Action.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

November 04, 2008